

MAXYME CHIASSON  
Projet de programmation  
420-P66-SI

Documentation finale  
Projet Drone et IA

Travail présenté à  
Martin Carignan

Département d'informatique  
Cégep de Sept-Îles  
Le 30 mai 2018

## Table des matières

Introduction.....	3
___ Le parcours.....	4
Préparation du code.....	6
___ Créer une app sur DJI.....	6
___ Télécharger le SDK de DJI.....	8
___ Télécharger le parcours .....	9
___ Compilation.....	11
___ Authentification .....	12
Exécution du code .....	14
___ Connexion SSH .....	15
Problèmes rencontrés .....	16
MÉDIAGRAPHIE .....	17

# Introduction

Ce présent document fait objet de guide pour le projet de drone et d'intelligence artificielle, dans le cadre du cours « Projet de programmation ». L'ensemble de la gestion du projet y est couvert, ainsi que la marche à suivre pour être en mesure d'exécuter le code afin de faire le parcours dans le gymnase. À la fin du document, vous trouverez une description des problèmes rencontrés lors de la réalisation de ce projet, ainsi qu'une médiagraphie contenant l'ensemble des sources utilisées tout au long de ce projet.

À la remise du drone par le client, beaucoup de recherche s'imposait. En consultant la documentation de DJI et leur site web pour développeur, on s'y retrouve tout de même aisément. Les « *samples*<sup>1</sup> » de DJI permettent de décortiquer le code quelque peu, et de savoir un peu par où commencer.

Pour arriver à faire le projet, notre drone devait faire trois parties bien précises et défini par le client, soit :

- Effectuer un parcours dans le gymnase, avec certaines contraintes
- Suivre une ligne rouge dans le gymnase grâce à la caméra embarqué du drone
- Rechercher et trouver une balle d'une couleur quelconque dans le gymnase

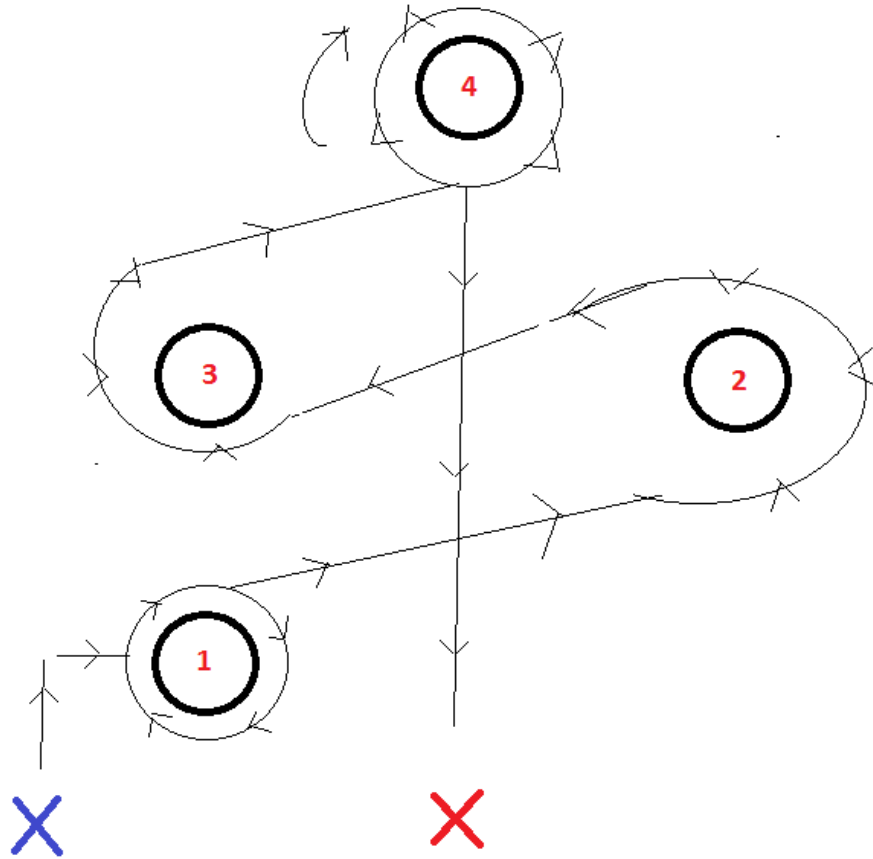
Étant donné qu'uniquement la première partie du projet fut remise, ce document traitera seulement de cette partie en question.

---

<sup>1</sup> Exemple de programme de DJI, disponible sur la partie développeur de leur site web.

## Le parcours

Le parcours consiste à faire décoller le drone, le faire tourner le drone autour de quatre cônes d'une façon bien précise et finalement, le faire atterrir. Ci-dessous, est illustré le parcours et la direction que le drone doit prendre, par rapport au nez du drone (et par le fait même, la position de la caméra).



Les cercles noirs représentent les cônes du parcours, et à l'intérieur ils sont numérotés en rouge dans l'ordre de passage. Le X bleu représente la position de départ de l'appareil, tandis que le rouge représente le point d'atterrissage. Le drone doit tourner autour des cônes de la façon suivante :

- 1- Tourne en pointant vers l'avant, pour former un cercle dans le sens horaire.
- 2- Tourne autour du cône, avec le nez pointant sur le cône lors du virage.
- 3- Tourne autour du cône, avec le nez pointant sur le cône lors du virage.
- 4- Tourne autour du cône, dans le sens horaire, formant un cercle et avec le nez pointant à l'extérieur de celui-ci.

Une fois dans le gymnase, le parcours prend la forme suivante :



## Préparation du code


La procédure à suivre pour envoyer le code sur l'ordinateur embarqué du drone se divise en plusieurs étapes.

Il est important de noter que ce guide ne porte pas sur la configuration de l'OSDK de DJI sur le drone, ni de la configuration du drone en tant que serveur SSH, ni de la configuration d'une clé Wi-Fi sur le Manifold. Nous considérerons pour les prochaines étapes que ces configurations sont déjà en place et qu'elles sont opérationnelles.

Vous noterez par contre la division en deux étapes : la première se veut une préparation avant le vol, et la deuxième expliquant comment faire exécuter le parcours. La première partie devrait être faite avec moniteur, clavier et souris de brancher sur le manifold. L'accès internet sur le manifold n'est pas nécessaire (quoi que toutefois recommandé), vous pouvez également transférer votre OSDK avec une clé USB par exemple, si vous l'aviez déjà téléchargé. La deuxième partie quant à elle, se fera dans le gymnase, donc vous aurez besoin d'un ordinateur portable avec un client SSH pour accéder aux commandes du drone à distance.

### Se créer une « app » sur DJI développeur

Afin d'exécuter le code, vous devez être authentifié auprès de DJI, sans quoi le drone ne s'initialisera pas, et refusera de décoller. Il est donc impératif de faire les bonnes manipulations afin de bien s'authentifier.

Pour ce faire, rendez-vous à [l'adresse suivante](#) pour vous connecter sur DJI Developer. Une fois authentifié, vous devrez vous créer une « app ». C'est simple, il suffit d'appuyer sur  et une fenêtre apparaît, nous invitant à entrer les informations nécessaires.

CREATE APP

SDK

Onboard SDK

APP Name

Parcours

Category

Other

Educational

Description

Parcours réalisé dans le gymnase du Cégep de Sept-Îles dans le cadre du cours 420-P66-SI du programme de Technique de l'informatique de gestion.

CANCEL

CREATE

Vous devez absolument prendre le « **Onboard SDK** », et définir un nom pour votre application. Pour le reste, c'est sans importance, libre à vous d'écrire ce que bon vous semble. Par contre, vous deviez être le plus précis possible, alors servez-vous de l'illustration ci-haute comme exemple. Une fois terminé, appuyez sur « **Create** », et DJI vous envoie une confirmation à votre courriel lié à votre compte. Vous devez ouvrir le lien dans ce courriel, afin de valider votre application, tel qu'illustré ci-bas.



Dear djiuser\_JuX0LWqSwZKx:

Thank you for your app creation, please click the activation link below to activate your app.

Activation link: [https://dev.dji.com/app/app\\_activate?token=a90c97e116761de9bc80ea0f3adeddb7&data=VOu1OctLsKdU5jTBeHX2LrlWPcqWnJcrmQ2npMdYH\\_D2dWTuHYcOiSI5qEPbuzp](https://dev.dji.com/app/app_activate?token=a90c97e116761de9bc80ea0f3adeddb7&data=VOu1OctLsKdU5jTBeHX2LrlWPcqWnJcrmQ2npMdYH_D2dWTuHYcOiSI5qEPbuzp)

Ce lien vous amène directement dans votre application, et vous affiche votre « **APP ID** » ainsi que votre « **APP Key** », tel qu'illustré ci-dessous.

#### APP INFORMATION


SDK Type	Onboard SDK
APP Name	Parcours
APP ID	<b>Votre APP ID</b>
App Key	<b>Votre APP Key</b>
Category	Educational
Description	Parcours réalisé dans le gymnase du Cégep de Sept-Îles dans le cadre du cours 420-P66-SI du programme de Technique de l'informatique de gestion.

 EDIT

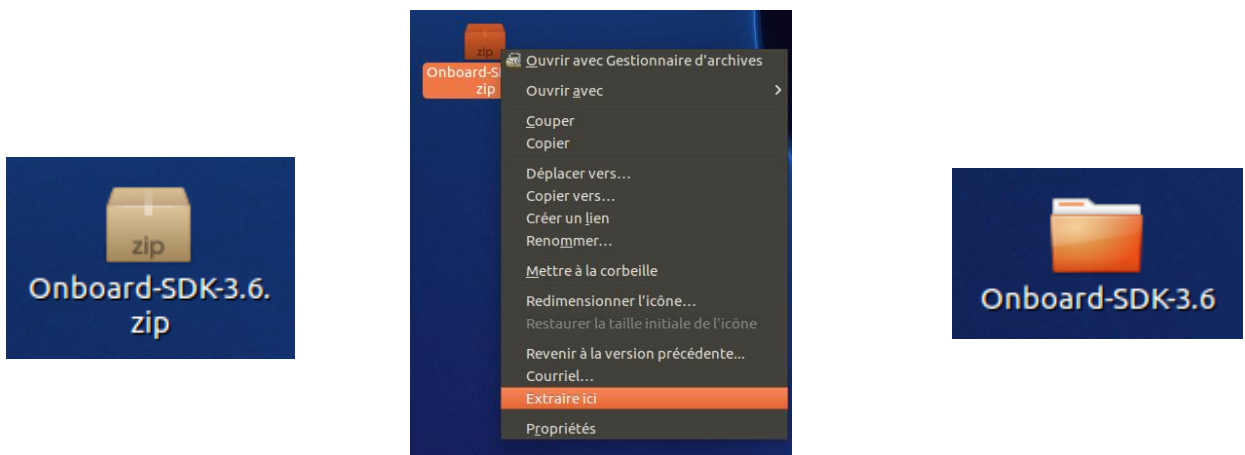
 DELETE

**Ces informations sont importantes pour la prochaine étape. Prenez-les en notes.**

## Télécharger le « Onboard SDK » de DJI

Rendez-vous à [l'adresse indiquée](#) pour y télécharger la dernière version de « **DJI Onboard SDK** » directement sur le Manifold. Une fois sur la page, appuyez sur  pour télécharger un fichier compressé (.zip) du projet.


Pour des principes de facilité, nous manipulerons le fichier directement sur le bureau du Manifold. Vous devez maintenant extraire l'archive en « .zip » que vous venez de télécharger. Il porte le nom « **Onboard-SDK-X.X** », où les **X** sont les valeurs que prenne le numéro de version. Pour ce faire, il suffit de faire un clic droit sur le fichier zip, et de cliquer sur « **Extraire ici...** ».



*\*Nous vous suggérons de ne pas supprimer le fichier zip, si jamais il arrivait malheur à votre SDK actuel, vous n'aurez pas besoin de le télécharger une nouvelle fois.*

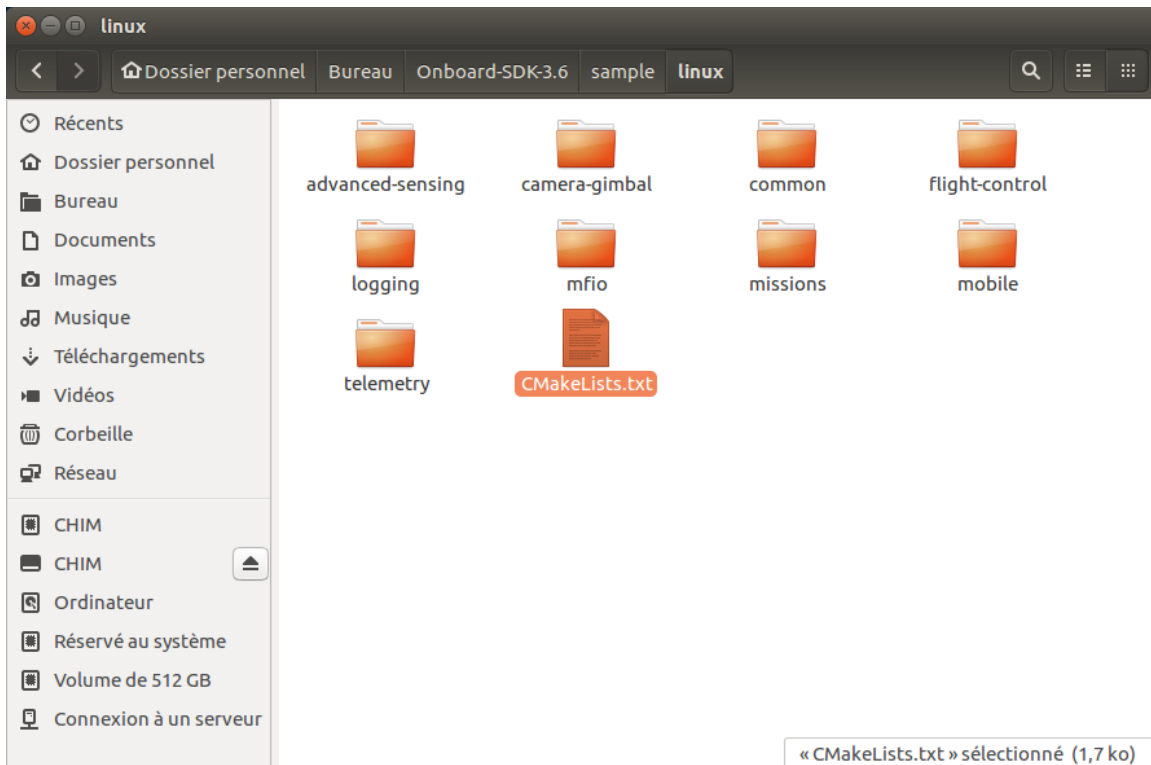


## Télécharger le parcours

Rendez-vous à [l'adresse indiquée](#) pour y télécharger le projet du parcours. Une fois sur la page, appuyez sur  pour télécharger un fichier compressé (.zip) du projet.

Il s'agit encore une fois de faire le même processus que pour le téléchargement précédent, on dépose le fichier à un endroit quelconque (le bureau dans notre cas) et on extrait l'archive.

Une fois le fichier extrait, déplacez ce dossier dans le répertoire « **/sample/linux** » de votre SDK.



Dans ce même répertoire, vous devrez apporter une légère modification au fichier « **CMakeLists.txt** », afin que votre SDK inclue le parcours dans sa compilation. Pour ce faire, ouvrez le fichier avec un éditeur de texte, tel qu'illustré à la page suivante.

```

24 cmake_minimum_required(VERSION 2.8)
25 project(onboardsdk-linux-samples)
26
27 if(NOT ONBOARDSDK_SOURCE)
28     set(ONBOARDSDK_SOURCE "${CMAKE_CURRENT_SOURCE_DIR}/../../osdk-core")
29 endif()
30
31 add_subdirectory(camera-gimbal)
32 add_subdirectory(flight-control)
33 add_subdirectory(missions)
34 add_subdirectory(telemetry)
35 add_subdirectory(logging)
36 add_subdirectory(parcours)
37
38 if(ADVANCED_SENSING)
39     add_definitions(-DADVANCED_SENSING)
40     message(STATUS "AdvancedSensing sample is enabled")
41     add_subdirectory(advanced-sensing)
42 endif()

```

Vous devrez rajouter une ligne de code, après les déclarations des « **add\_subdirectory(project\_name)** ». Faites un saut de ligne, et rajouté la ligne suivante :

```
add_subdirectory(parcours)
```

Sauvegarder et quitter votre éditeur de texte.

## Compilation

Nous en sommes maintenant à la compilation. Pour ce faire, ouvrez un terminal sur le manifold et accédez à votre SDK (Sur le bureau dans notre cas. Votre commande peut différer selon l'emplacement de votre SDK.).

```
max@CHIM:~$ cd /home/max/Desktop/Onboard-SDK-3.6
```

Ensuite, vous devrez créer un dossier nommé « **build** » dans votre dossier. Placez-vous ensuite dans ce dossier. Ensuite nous utiliserons les commandes de **CMAKE** pour compiler. Entrez donc les commandes suivantes

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
make
```

L'affichage de votre terminal devrait ressembler de près à ceci :

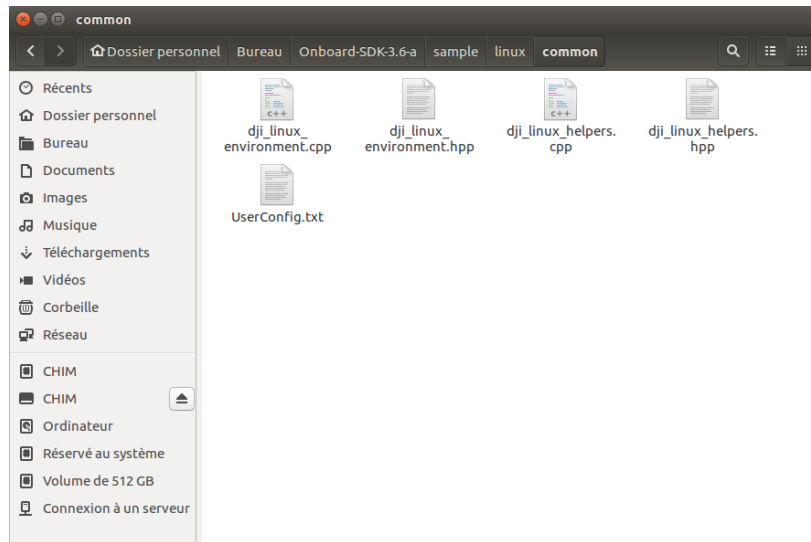
```
max@CHIM: ~/Bureau/Onboard-SDK-3.6/build
max@CHIM:~/Bureau/Onboard-SDK-3.6$ mkdir build
max@CHIM:~/Bureau/Onboard-SDK-3.6$ cd build
max@CHIM:~/Bureau/Onboard-SDK-3.6/build$ cmake ..
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/max/Bureau/Onboard-SDK-3.6/build
max@CHIM:~/Bureau/Onboard-SDK-3.6/build$
```

```
max@CHIM:~/Bureau/Onboard-SDK-3.6/build$ make
```

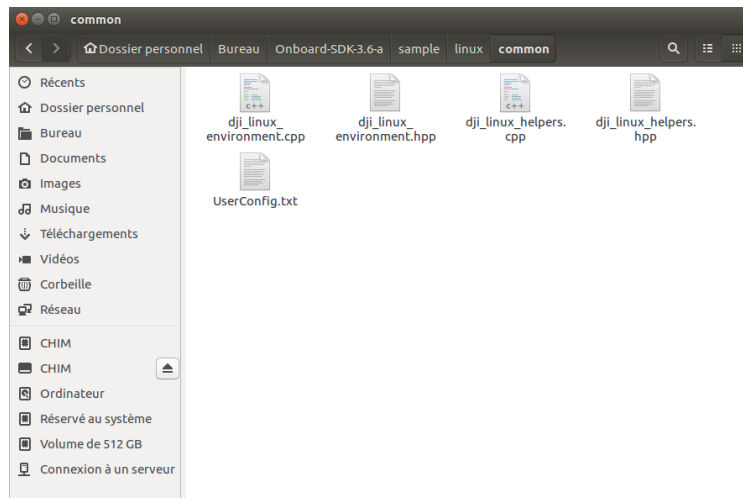
Ces commandes ne devraient pas vous donner d'erreurs. Si c'est le cas, pensez à vous référer à la section « **Problèmes rencontrés** » de ce document.

## Authentification

C'est ici que vous configurez votre application. La première étape est de vous rendre dans le répertoire « **/sample/linux/common** » de votre SDK. Dans ce répertoire, se trouve un fichier nommé « **UserConfig.txt** ».



Copier ce fichier dans le répertoire « **/build/bin** » de votre SDK. Ouvrez ensuite le fichier avec un éditeur de texte.



```
1 app_id : Votre app_id
2 app_key : Votre app_key
3 device : /dev/ttyTHS1
4 baudrate : 230400
```

Vous devez donc mettre les informations nécessaires pour l'activation de votre drone. Les deux premiers champs correspondent à vos numéros d'identification et votre clé d'application, que vous avez créé durant les étapes précédentes. Les deux derniers champs devraient être identiques à l'illustration.

Vous pouvez également les remplacer par ces lignes directement :

**device:/dev/ttyTHS1**

**baudrate : 230400**

Sauvegarder le fichier, et préparez-vous à aller dans le gymnase.

## Exécution du code

La première étape avant de voler, est de faire la routine de pré-décollage. Assurez-vous que tous les équipements sur la Matrice 100 sont bien fixés, que la batterie est au minimum à plus de 50 % de sa capacité (dans le cas contraire, remplacez-là par une pleine.), et que l'environnement de vol est bien dégagé (panier de basketball relevé, cordes du mur d'escalade rangées, etc.).

Branchez votre routeur et assurez-vous que le manifold est bel et bien connecté dessus. Branchez l'ordinateur portable sur le même réseau. Si votre adresse IP n'est pas statique, vous devez la consulter avant de débrancher le moniteur, puisqu'elle est nécessaire pour la connexion par SSH. Pour ce faire, ouvrez un terminal et tapez la commande suivante :

```
ifconfig
```

Cette commande vous affichera les sorties suivantes :

```
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:13:14:d5
        inet addr:192.168.1.160  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::26ab:3be8:ef5f:41bf/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:41 errors:0 dropped:0 overruns:0 frame:0
        TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:6971 (6.9 KB)  TX bytes:7517 (7.5 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:1727 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1727 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:107853 (107.8 KB)  TX bytes:107853 (107.8 KB)
```

Selon l'interface, vous aurez l'adresse que vous avez besoin. Prenez cette adresse en note. Débranchez maintenant le moniteur, et placez votre drone sur son point de départ.

Pour le restant de ce guide, nous considérerons l'adresse de notre drone comme étant « **192.168.1.50** », à titre d'exemple.

Faites une dernière vérification des équipements et attachez les hélices si ce n'est pas déjà fait. Ouvrez la manette, connectez-y un appareil mobile exécutant l'application « **DJI GO** » et mettez la manette en mode « **F** ».

## Connexion SSH

La prochaine et dernière étape, est de vous connecter en SSH sur votre drone. Pour ce faire, ouvrez votre terminal et tapez la commande suivante :

```
ssh ubuntu@192.168.1.50
```

Le terminal vous invite ensuite à entrer le mot de passe du compte Ubuntu du Manifold. À l'heure de la production de ce document, le mot de passe était « **ubuntu** ».

Une fois la connexion établie, naviguer dans votre SDK, puis dans le répertoire « **/build/bin** ». C'est dans ce dossier que se trouve l'exécutable de notre parcours. Entrez donc les commandes suivantes :

```
cd / Desktop/ Onboard-SDK-3.6/ build/ bin
```

```
./parcours
```

Et le parcours devrait commencer. Il se peut que le terminal vous donne une erreur d'authentification la première fois. Si cela arrive, refaites la dernière commande et cela devrait passer.

Le drone partira ses moteurs, fera son décollage, suivra le parcours, puis atterrira.

## Problèmes rencontrés

Plusieurs problèmes ont été rencontrés lors de la réalisation de ce projet. En voici quelques-uns :

- Accès ultra limité au drone pendant les quatre premiers mois de la session.
- Seulement un drone pour six personnes effectuant le même projet, jusqu'à l'achat d'un deuxième drone, un mois avant la remise du projet.
- Impossible d'avoir accès au drone en dehors des heures du collège, jusqu'à un mois et demi avant la remise du projet.
- Très peu de théorie sur le fonctionnement du drone
- Énormément d'heures d'absences dans les cours
- Utilisez des librairies inconnues.
- Difficile d'ajuster les angles du parcours parfaitement.
- Influence de la climatisation du gymnase sur le drone durant le vol

Les étudiants (nous), ont fait des démarches pour avoir un accès au drone plus important, sans quoi, personne ne pouvait avancer convenablement son projet.

La seule documentation présente est sur le site de DJI développeur, où il n'est pas toujours facile de s'y retrouver.

Il ne va pas s'en dire que les nombres heures d'absences de ma part dans ce cours, on grandement contribué à mes difficultés dans ce projet, en plus de prendre du retard dans le projet.

Une des difficultés au niveau de la programmation, fut de se familiariser avec des librairies en C++ qui m'était entièrement inconnues.

Les deux derniers points sont dans les plus importants. La différence sur le simulateur versus dans le gymnase est assez important, étant donné la force des courants causée par le système de climatisation. Celui-ci fait parfois dévier l'aéronef de sa trajectoire, et change les mesures et les angles en cours de route. Il faut donc s'ajuster dans le gymnase directement. Chose que je n'ai pas pu faire puisque je n'ai jamais testé mon parcours dans le gymnase avant sa remise.



# MÉDIAGRAPHIE

- 1- **DJI Developer**, Authentication, 2017, 22 mars 2018, <https://developer.dji.com/login/>
- 2- **DJI Developer**, Setting up Samples, 2017, 23 mars 2018, <https://developer.dji.com/onboard-sdk/documentation/sample-doc/sample-setup.html#linux-onboard-computer>
- 3- **DJI Developer**, DJI Onboard SDK Documentation, 2017, 6 avril 2018, <https://developer.dji.com/onboard-api-reference/index.html>
- 4- **GitHub**, DJI-SDK, Onboard-SDK, 2018, 6 avril 2018, <https://github.com/dji-sdk/Onboard-SDK>
- 5- **DJI Developer**, Control Class Reference, 2017, 6 avril 2018, [https://developer.dji.com/onboard-api-reference/classDJI\\_1\\_1OSDK\\_1\\_1Control.html](https://developer.dji.com/onboard-api-reference/classDJI_1_1OSDK_1_1Control.html)
- 6- **DJI Developer**, CtrlData Struct Refence, 2017, 6 avril 2018, [https://developer.dji.com/onboard-api-reference/structDJI\\_1\\_1OSDK\\_1\\_1Control\\_1\\_1CtrlData.html](https://developer.dji.com/onboard-api-reference/structDJI_1_1OSDK_1_1Control_1_1CtrlData.html)
- 7- **DJI Developer**, Onboard SDK Overview, 2017, 5 mars 2018, <https://developer.dji.com/onboard-sdk/>

